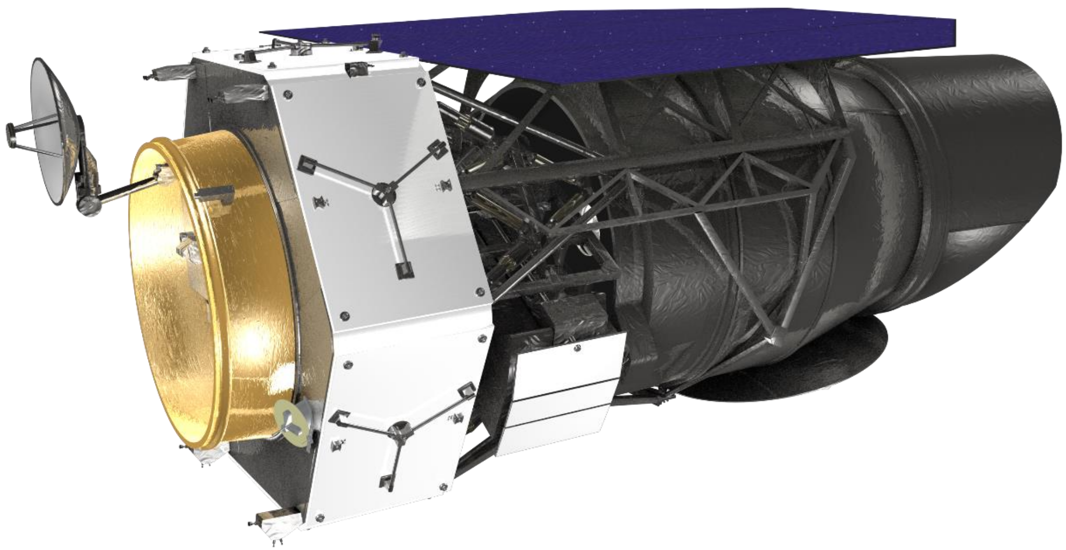
IMPipeline: An Integrated STOP modeling pipeline for the WFIRST coronagraph

SPIE Optics + Photonics 2017
Techniques and Instrumentation for Detection of Exoplanets VIII
August 8 2017



Navtej Saini[a], Kevin Anderson[a], Zensheu Chang[a], Gary Gutt[a], Bijan Nemati[a,b]

a) Jet Propulsion Laboratory, California Institute of Technology

b) The University of Alabama, Huntsville, AL

Talk Outline

1. Introduction
2. Fundamentals of STOP Analysis
3. Integrated Modeling Pipeline
4. Pipeline Automation
5. Future Work

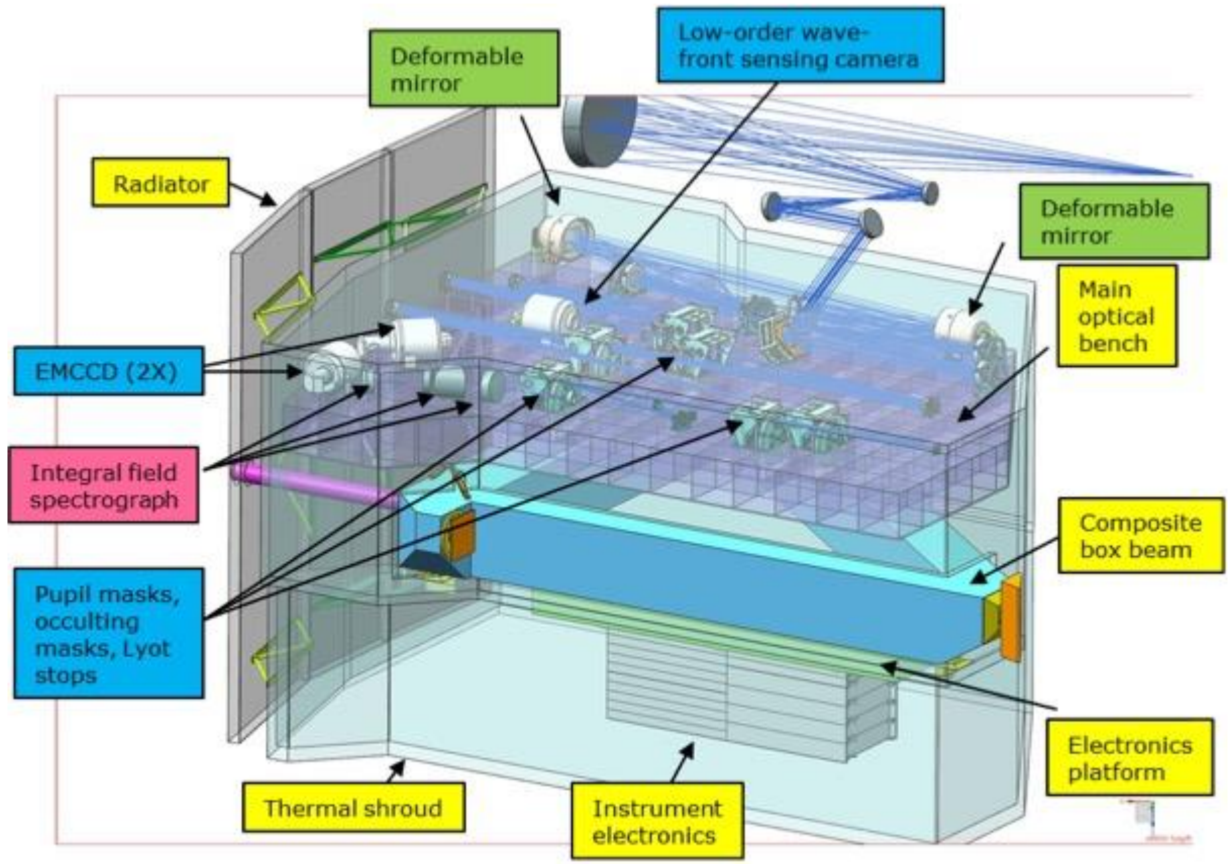
Wide Field Infrared Survey Telescope

WFIRST, the **W**ide **F**ield **I**nfra**R**ed **S**urvey **T**elescope, is a future NASA telescope to explore exoplanets and answer essential questions in the field of dark Energy and infrared astrophysics.

- 2.4 meter diameter primary mirror
- Designed for 6 year mission
- Will be located at the Earth-Sun Lagrange Point 2
- Two instruments onboard – Wide Field imager (WFI) and a Coronagraph (CGI)
- WFI will provide capabilities to perform the Dark Energy, Exoplanet microlensing, and NIR surveys
- CGI supports the Exoplanet high contrast imaging and spectroscopy science

Credit: NASA GSFC (https://wfirst.gsfc.nasa.gov/gallery-full.html)

WFIRST Coronagraph



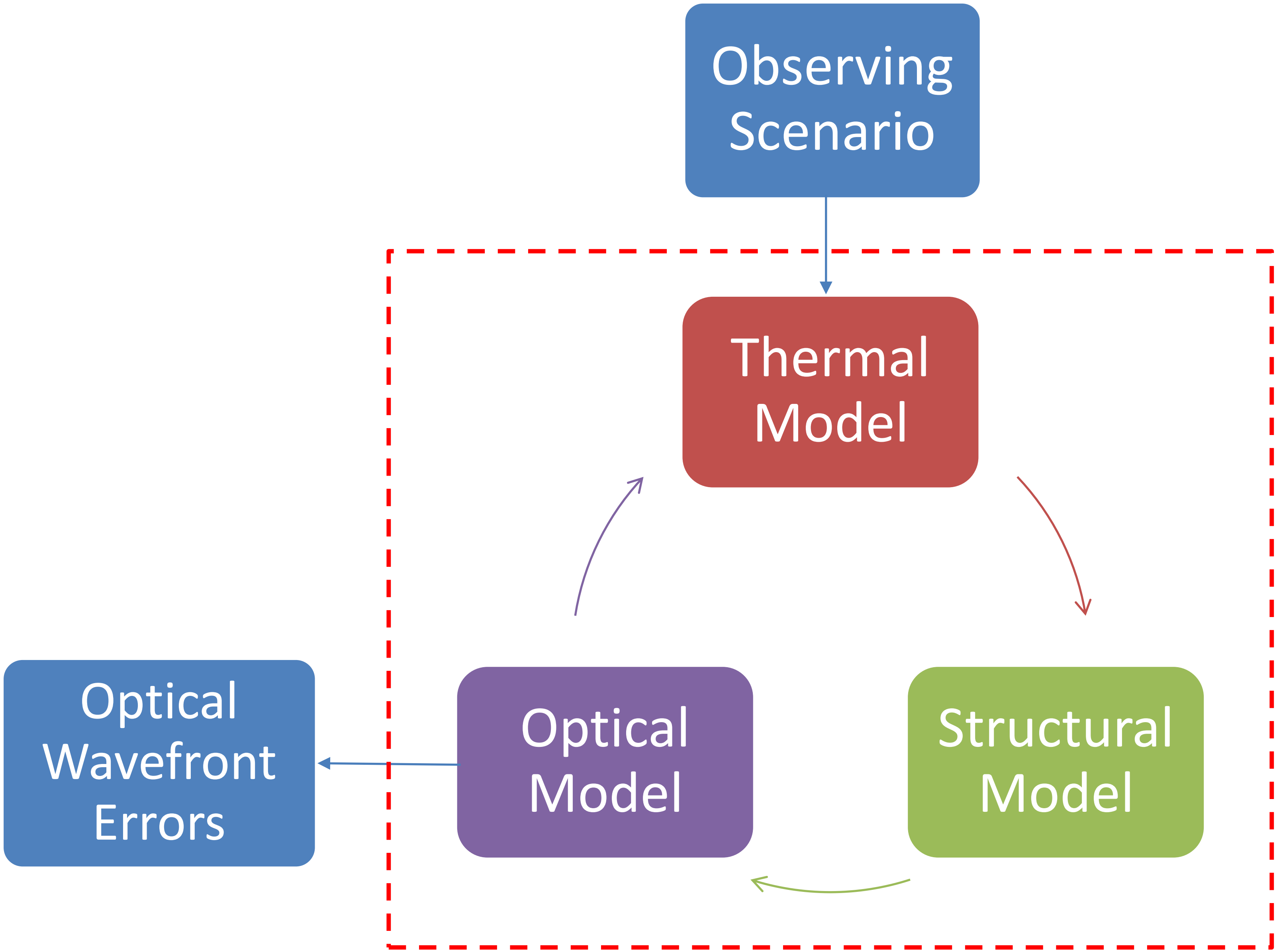
- Coronagraph imager covers spectral range of 0.43 – 0.98 µm
- Two deformable mirrors form a sequential wavefront control system (WFCS) that compensates for both phase and amplitude errors in the telescope and coronagraph optics
- The spectroscopy mode uses and integral field spectrograph (IFS)

Image credit: NASA GSFC (https://wfirst.gsfc.nasa.gov/observatory.html)

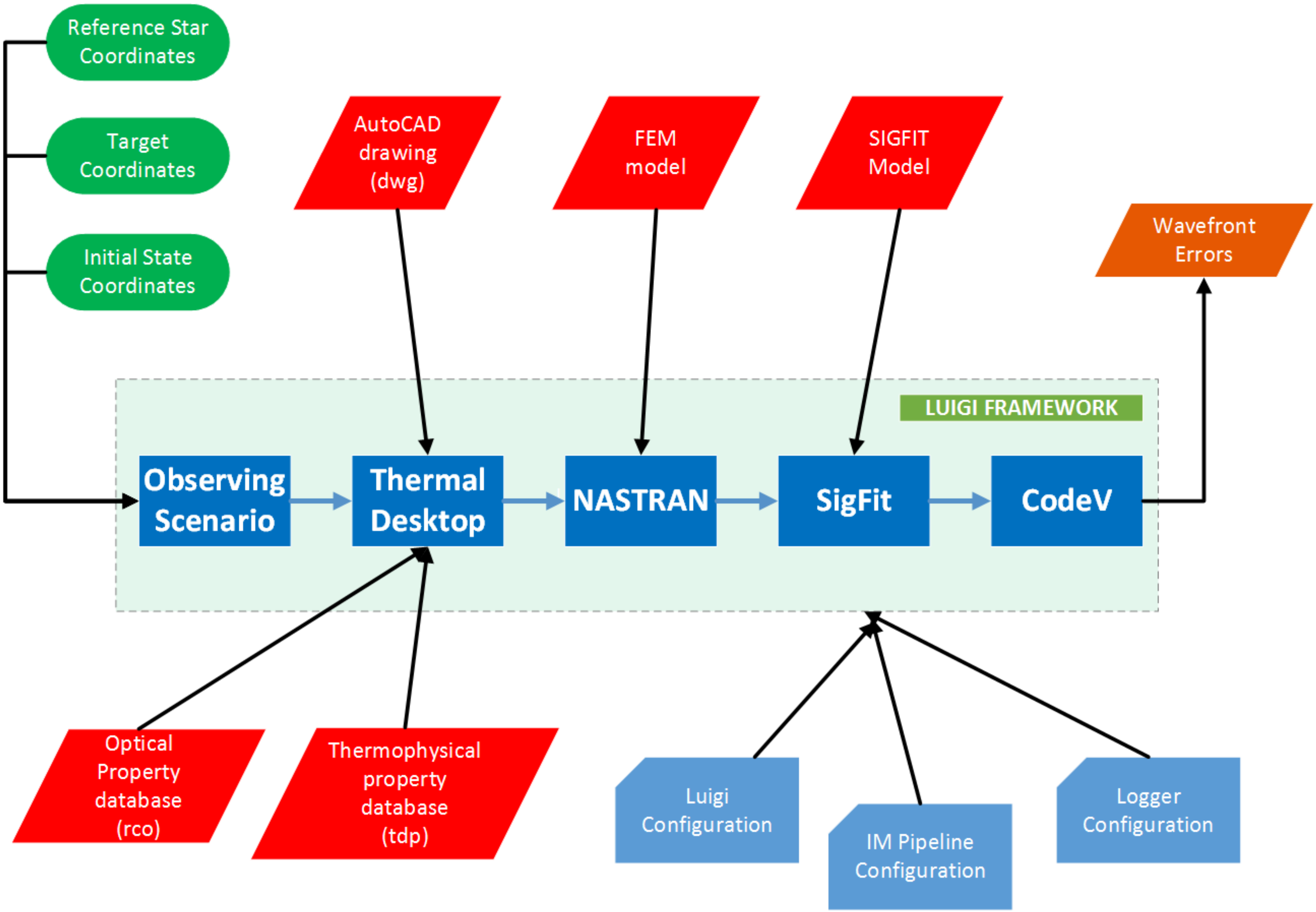
Fundamentals of STOP Analysis

- STOP analysis is a multidiscipline analysis, consisting of **S**tructural, **T**hermal, and **O**ptical **P**erformance analyses of space instruments under thermal load
- STOP analysis estimates the thermally-induced performance degradation of an optical system
- Thermal deformation is one of the major sources that can degrade the performance of a space telescope or antenna
- Due to the requirement for high accuracy, temperature dependent CTE is used for calculating thermal loads
- Wavefront errors due to optics motions (rigid-body motion) can be predicted from the STOP analysis
- Thermal loads are driven by environmental (orbital mechanics) conditions, the thermal control system must ensure that the componentry such as the Coronagraph Instrument (CGI) hold requirements on set-point with ample stability

IMPipeline: Motivation



IMPipeline Block Diagram



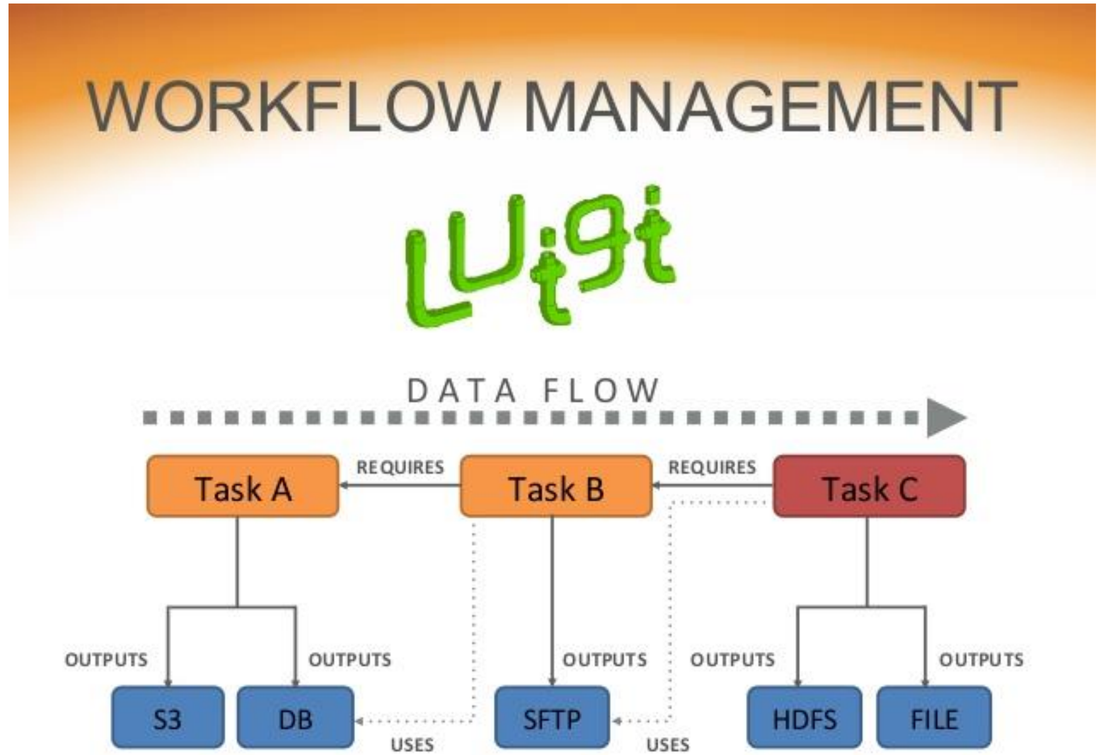
Pipeline Framework: Luigi

World's second most famous plumber

Source code:
https://github.com/spotify/luigi

License:
https://github.com/spotify/luigi/blob/master/LICENSE

Image credit:
Spotify Luigi



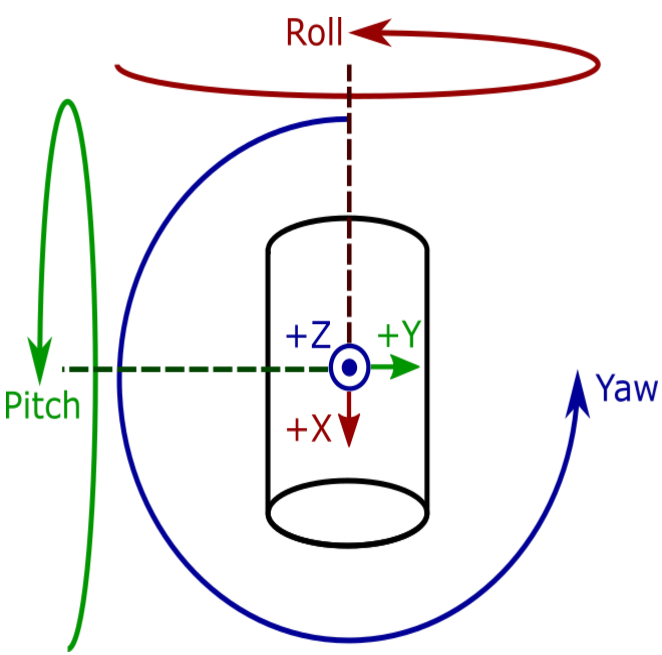
- Python framework to build complex pipelines of batch jobs
- Developed by engineers at Spotify and made open source for wider community
- Luigi server comes with a web interface to visualize, search and filter tasks
- Luigi is customizable, and handles dependency resolution, workflow management, visualization etc.
- Open source software licensed under Apache License 2.0

Observing Scenario

- Solar heating is a major input to WFIRST coronagraph thermal model
- The thermal model treats Solar radiation as plane wave incident on the spacecraft. We have to determine direction of the plane wave and the intensity of the flux
- During operations, the spacecraft attitude will be a function of the stars WFIRST will be observing, and the time of year during which they are observed.
- To simulate thermal effects of an observing scenario, we need to specify relation between the WFIRST spacecraft and the Sun.
- As WFIRST is expected to be at the Sun-Earth Lagrange point, Earth’s infrared radiation is neglected.

Observing Scenario cont'd

Solar vector X, Y, and Z components comprise a unit vector pointing in spacecraft body-fixed coordinates to the center of the Sun.



Spacecraft coordinate frame and rotations (William Schulze)

Heating Rate Case: OS5_L2_VECTOR

Vector Input | Planetary Data | Solar | Albedo | IR Planetshine | Fast Spin

Time [sec]	Solar Vector	Planet Vector	
0	0.3211, 0.0000, 0.9470	0.0000, 0.0000, -1.0000	1.050000
289999	0.3211, 0.0000, 0.9470	0.0000, 0.0000, -1.0000	1.050000
290000	0.2433, 0.0567, 0.9683	0.0000, 0.0000, -1.0000	1.050000
290199	0.2433, 0.0567, 0.9683	0.0000, 0.0000, -1.0000	1.050000
290200	0.1638, 0.1132, 0.9800	0.0000, 0.0000, -1.0000	1.050000
290399	0.1638, 0.1132, 0.9800	0.0000, 0.0000, -1.0000	1.050000
290400	0.0833, 0.1693, 0.9820	0.0000, 0.0000, -1.0000	1.050000
290599	0.0833, 0.1693, 0.9820	0.0000, 0.0000, -1.0000	1.050000
290600	0.0023, 0.2250, 0.9744	0.0000, 0.0000, -1.0000	1.050000
320599	0.0023, 0.2250, 0.9744	0.0000, 0.0000, -1.0000	1.050000
320600	0.0384, 0.2249, 0.9736	0.0000, 0.0000, -1.0000	1.050000
320799	0.0384, 0.2249, 0.9736	0.0000, 0.0000, -1.0000	1.050000
320800	0.0744, 0.2249, 0.9715	0.0000, 0.0000, -1.0000	1.050000
320999	0.0744, 0.2249, 0.9715	0.0000, 0.0000, -1.0000	1.050000
321000	0.1102, 0.2249, 0.9681	0.0000, 0.0000, -1.0000	1.050000
321199	0.1102, 0.2249, 0.9681	0.0000, 0.0000, -1.0000	1.050000
321200	0.1460, 0.2248, 0.9634	0.0000, 0.0000, -1.0000	1.050000
371199	0.1460, 0.2248, 0.9634	0.0000, 0.0000, -1.0000	1.050000
371200	0.1460, 0.1507, 0.9777	0.0000, 0.0000, -1.0000	1.050000
371219	0.1460, 0.1507, 0.9777	0.0000, 0.0000, -1.0000	1.050000
371220	0.1461, 0.0755, 0.9864	0.0000, 0.0000, -1.0000	1.050000
371239	0.1461, 0.0755, 0.9864	0.0000, 0.0000, -1.0000	1.050000
371240	0.1461, 0.0000, 0.9893	0.0000, 0.0000, -1.0000	1.050000
371259	0.1461, 0.0000, 0.9893	0.0000, 0.0000, -1.0000	1.050000
371260	0.1461, -0.0755, 0.9864	0.0000, 0.0000, -1.0000	1.050000
371279	0.1461, -0.0755, 0.9864	0.0000, 0.0000, -1.0000	1.050000
371280	0.1460, -0.1507, 0.9777	0.0000, 0.0000, -1.0000	1.050000
371299	0.1460, -0.1507, 0.9777	0.0000, 0.0000, -1.0000	1.050000
371300	0.1460, -0.2248, 0.9634	0.0000, 0.0000, -1.0000	1.050000
421300	0.1460, -0.2248, 0.9634	0.0000, 0.0000, -1.0000	1.050000

Solar Vector X = sin(pitch)

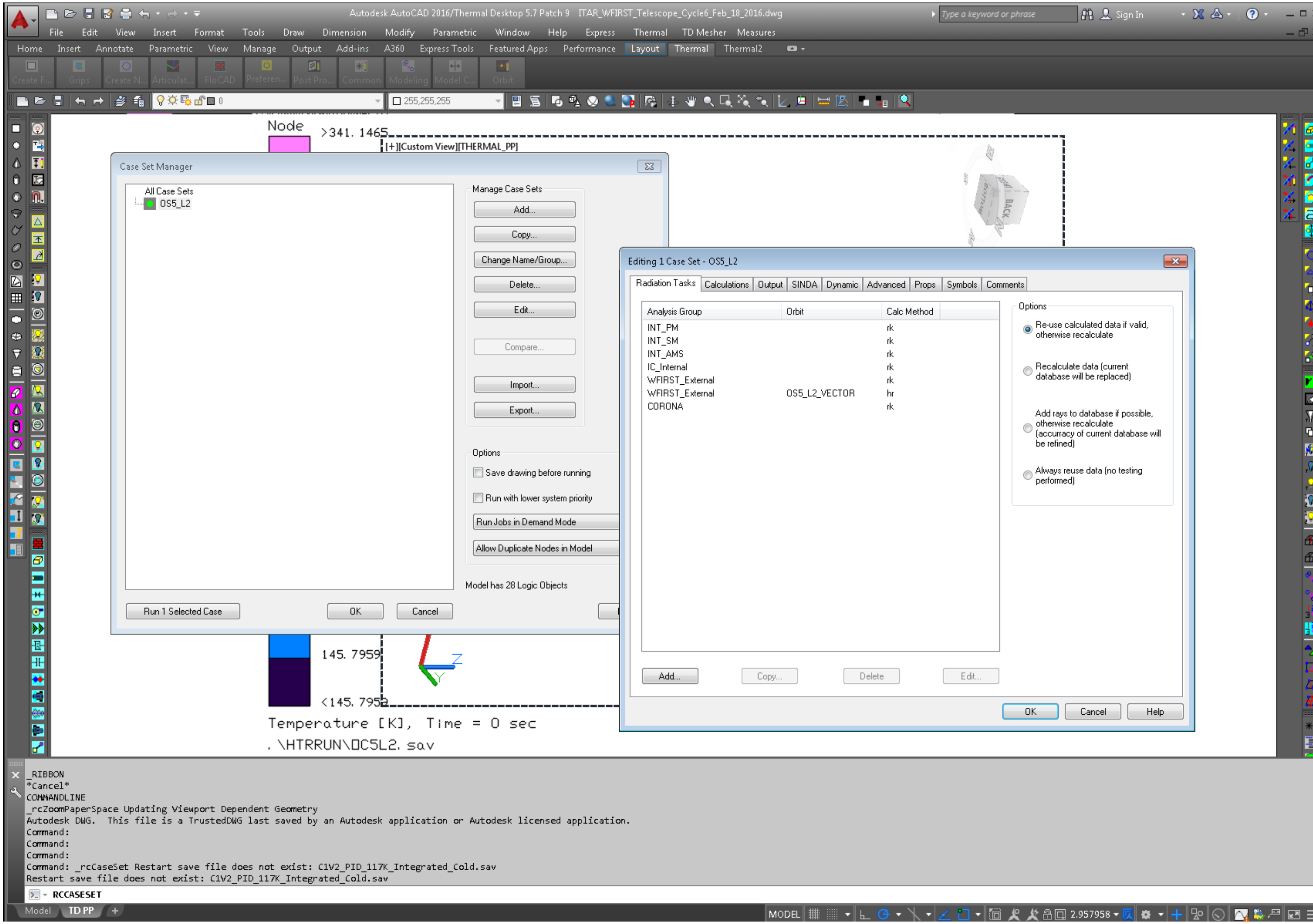
Solar Vector Y = cos(pitch) * sin(roll)

Solar Vector Z = cos(pitch) * cos(roll)

Thermal Modeling

- Thermal Desktop version 5.8 is used for thermal modeling
- Only available for Windows OS and is integrated with AutoCAD. User interaction is through AutoCAD GUI
- It creates the node and conduction network, launches SINDA/FLUINT for the solution, and provides post processing results
- It interfaces with high level programming languages using Component Object Model (COM). Component Object Model (COM) is a binary interface standard for software components. It is used to enable inter-process communication.
- We used Python’s **pywin32** package to interface with Thermal Desktop.

Thermal Modeling cont'd



Structural Modeling

- NX NASTRAN is a Finite Element solver for stress, vibration, buckling, structural failure, heat transfer, acoustic and aeroelasticity analyses.
- Structural FEM in FEMAP 10.2 format is used to map nodes between thermal and structural models
- Not all FEM nodes are mapped to the thermal model within the tolerance limit
- NASTRAN conduction analysis is used to fill-in the unmapped temperatures via interpolation
- Two step process – first step fill-in the unmapped temperatures and second step calculates the displacement of optical elements
- NASTRAN can be run from GUI as well as command line interface. Support exists for utilizing multiple processors/cores.

Optical Modeling

- Structural rigid body displacements cannot be directly imported to optical design software

- SigFit converts the displacements from NASTRAN to a format (Zernike polynomial, Interferogram etc.) that can be digested by the optical design software

- The SigFit output is used by CodeV with an existing optical model to generate wavefront errors (WFE)

Pipeline Execution

- Command line execution from terminal

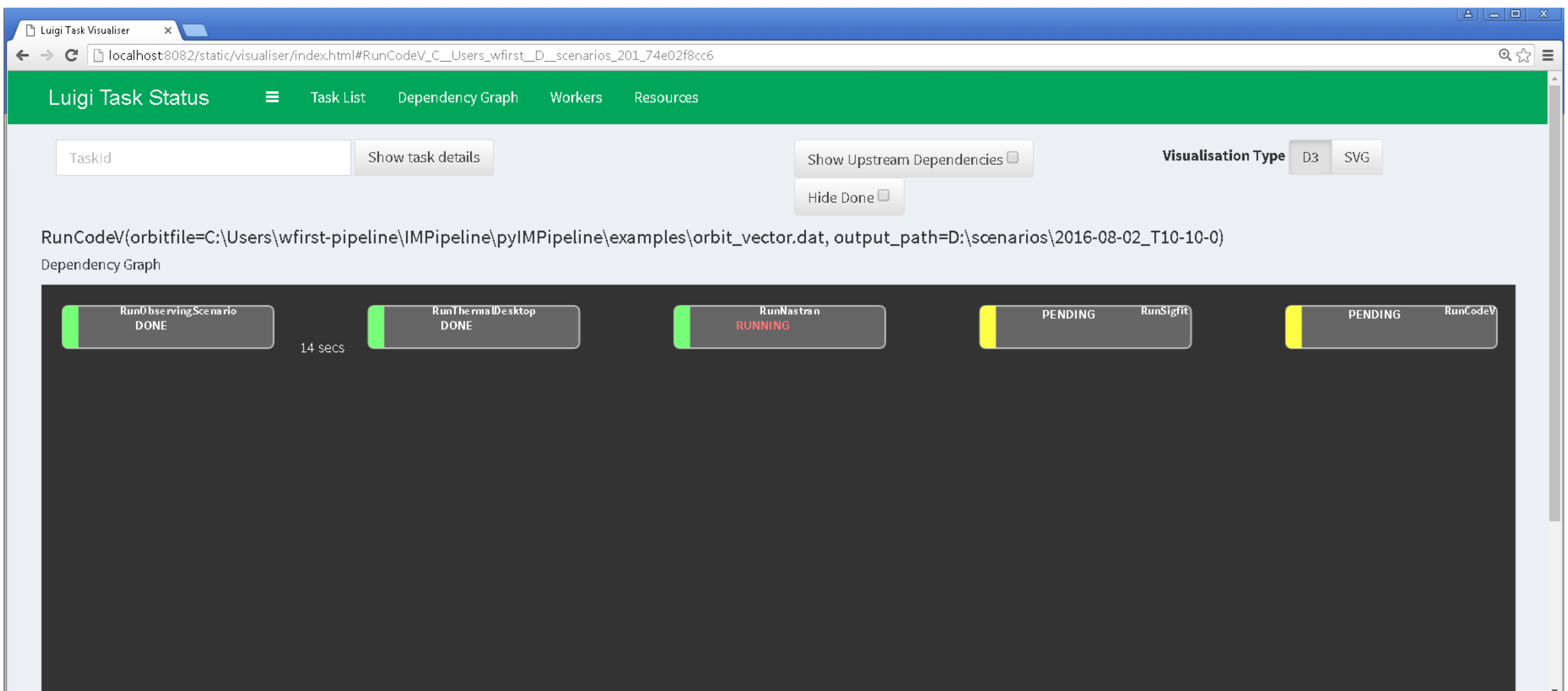
  > python impipeline.py RunCodeV --obsdate YYYYMMDD --ref-ra XXX.XXX --ref-dec XX.XXX --target-ra XXX.XXX --target-dec XX.XXX --initial-ra XXX.XXX --initial-dec XX.XXX --outpath XXXXXXXXX

- Command line input parameters:
  - Mandatory
    - --obsdate : observing date in YYYYMMDD format
    - --ref-ra : RA of the reference star
    - --ref-dec : DEC of the reference star
    - --target-ra : RA of the target
    - --target-dec : DEC of the target
  - Optional
    - --initial-ra : RA of initial pointing
    - --initial-dec : DEC of initial pointing
    - --outpath : Output directory where outputs are written to

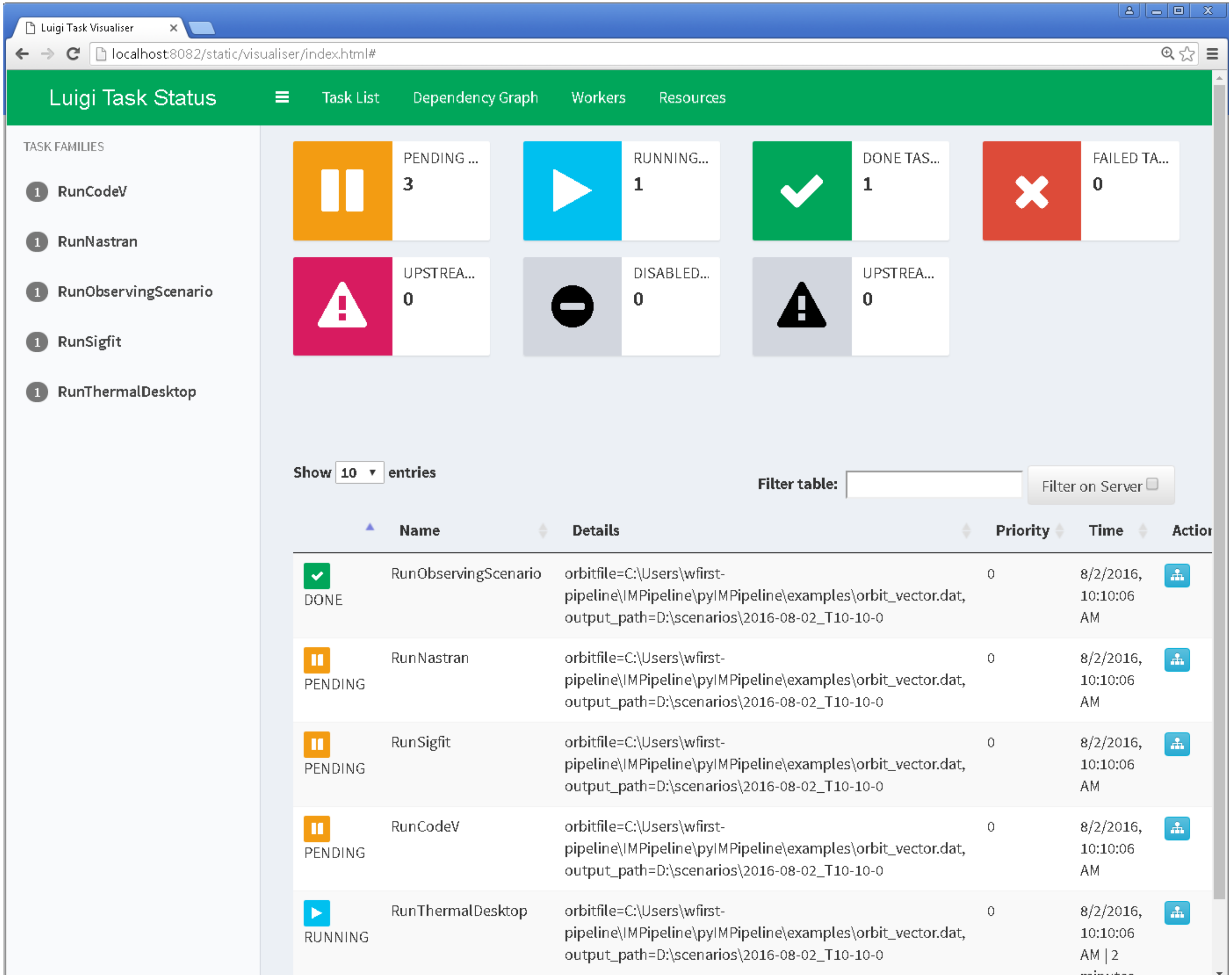
Pipeline Visualization

- Pipeline execution status can be checked using Luigi's web interface. Luigi runs a web server (Tornado) in the backend and displays task status in the web interface.

- Web interface can be accessed by going to following URL in web browser:
  http://localhost:8082/static/visualiser/index.html



Pipeline Visualization cont'd



WFIRST Coronagraph

Pipeline Output & Diagnostics

- Wavefront errors (WFE) is the main performance matrix coming out of the pipeline
- All the Thermal desktop, NASTRAN, SigFit and CodeV output files are saved for diagnostic purpose
- Diagnostics plots are generated at each step. Makes it easy to track down issues for individual models
- A consolidated log file (impipeline.log) is generated for each run. It includes messages from all the tasks, along with timestamps. It can be used to track down errors and warnings
- Using Luigi check pointing, it is easy to re-run the pipeline from any intermediate task

Customizing Pipeline

- IMPipeline can be easily customized using configuration files
- Three configuration files –
  - impipeline.cfg : configure various tasks in the pipeline
  - luigi.cfg : configure Luigi framework itself
  - logger.cfg : configure logging capability of the pipeline
- Email notification. Sends out an automated email notification if the pipeline fails with an error.
- User level customization. Pipeline is developed to be run by multiple people involved in the project. The configuration files allows users to customize the pipeline run based on their individual requirements.

Pipeline Infrastructure

- A dedicated high end rack server with enough firepower to process large thermal and structural models
- Windows 7 Professional operating system – due to licensing issues, Thermal Desktop runs only on Windows desktop editions
- Hardware details –
    - Processor : Intel Xeon @ 2.6 GHz, 10 core, 20 hyper threads (2 processors)
    - 256 GB RAM but only 192 GB useable
    - 30TB RAID storage
- Volume license for Thermal Desktop, NASTRAN, SigFit and Code V

Future Work

- Validation of CGI performance using updated observatory STOP model (Phase A)
- Create a library of wavefront errors for various “Operational Scenarios”
- Extend the IMPipeline beyond STOP analysis by Integrating the CGI wavefront propagation code in the pipeline to generate realistic speckle patterns
- A graphical user interface (GUI) for the pipeline for better usability

Thank You!

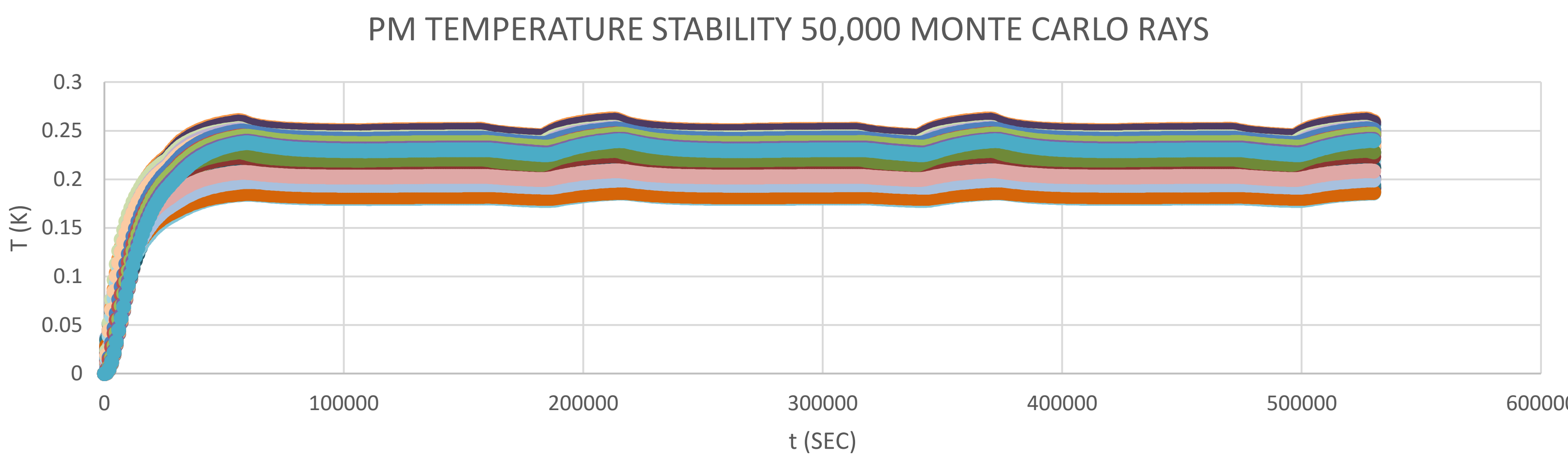
Backup Slides

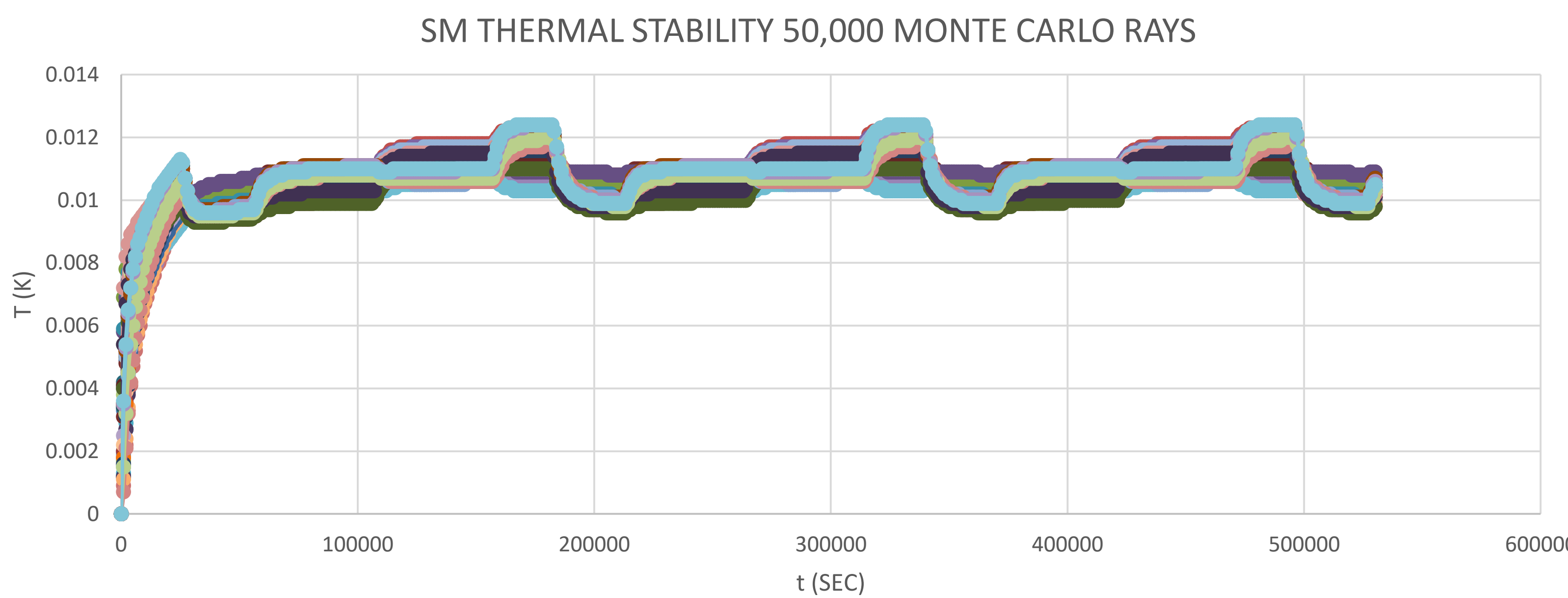
Observing Scenario – Pseudocode

- Determine position of Sun relative to Earth in ecliptic coordinate system
- Check if the target is within the Sun exclusion angle
- Transform position of reference star and target to ecliptic coordinate system
- Express all positions in heliocentric ecliptic coordinates
- Define spacecraft body centered coordinate system (BCCS)
- Determine transformation matrix between heliocentric coordinates and BCCS
- Initial orientation of spacecraft is +Z pointing towards the Sun and +X (boresight) direction pointing south out of the equatorial
- Slew to reference star
- Slew to target 2 and repeat steps as for target 1
- Reference Star Differential Imaging (RDI) roll
  - Roll about X for specified angle (+13 degrees to -13 degrees)
- Generate a text file with instantaneous time and Solar vector angles for telescope maneuver from initial state to target star

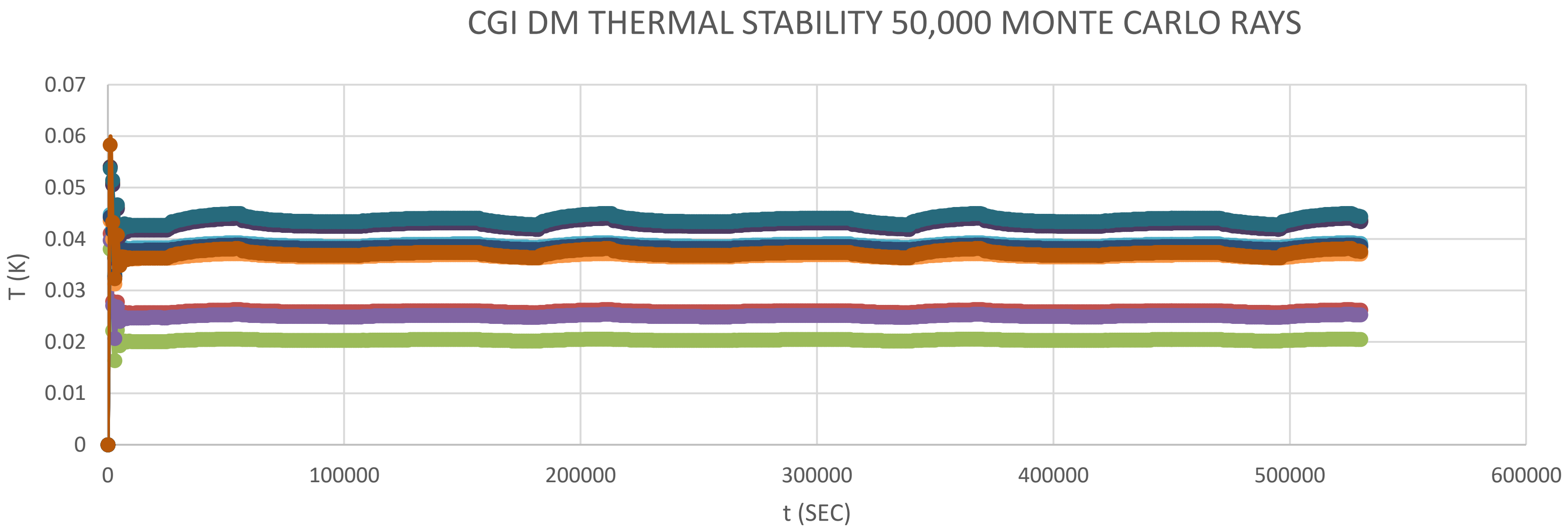
Preliminary Results - I

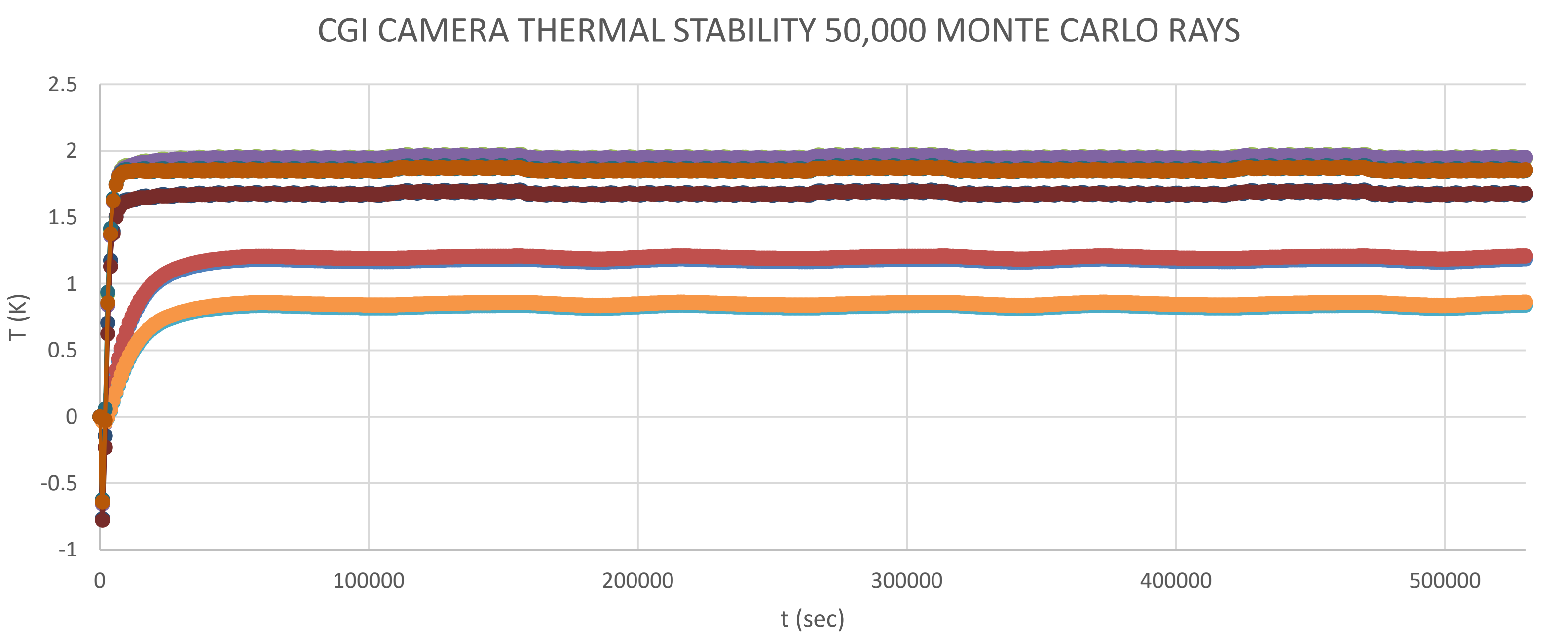
Exercise pipeline to investigate thermal stability of the instrument





Preliminary Results - II





Monte Carlo Ray Tracing Trade Study

A trade study was performed to determine the optimal # of rays per node versus error and the associated runtime